PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 :
G01S 5/18, 5/30, 15/87

A1

(11) International Publication Number: WO 99/64887

(43) International Publication Date: 16 December 1999 (16.12.99)

(21) International Application Number: PCT/GB99/01858

(22) International Filing Date: 11 June 1999 (11.06.99)

(30) Priority Data:
9812635.2 11 June 1998 (11.06.98) GB

(71) Applicant (*for all designated States except US*): AT & T LABORATORIES CAMBRIDGE LIMITED [GB/GB]; 24a Trumpington Street, Cambridge CB2 1QA (GB).

(72) Inventor; and
(75) Inventor/Applicant (*for US only*): STEGGLES, Peter, Joseph [GB/GB]; 70 Melvyn Way, Histon, Cambridge CB4 9HZ (GB).

(74) Agent: ROBSON, Aidan, John; Reddie & Grose, 16 Theobalds Road, London WC1X 8PL (GB).

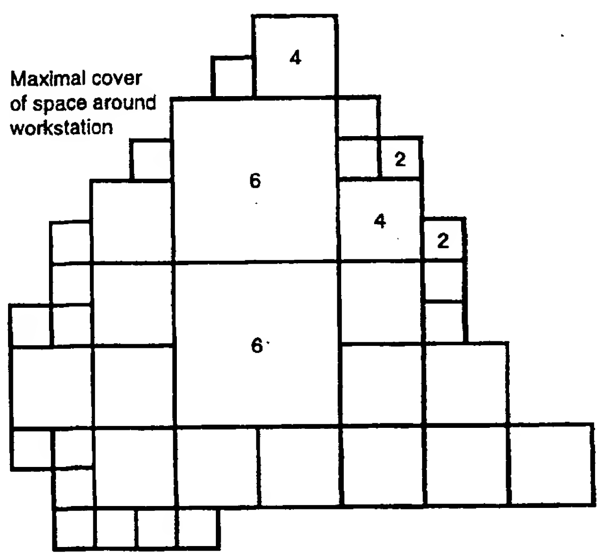
(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, HU, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

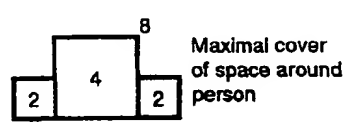
**Published**
*With international search report.*

(54) Title: LOCATION SYSTEM

(57) Abstract

A location system for objects having location devices (16) as a means for repeatedly interrogating the location devices on each object to provide position signals. A means (18) to detect the position signals are provided. The position of each object and the area associated with each object are determined from the position signals and stored. The system is then able to determine the relative locations of the thus determined objects and associated areas.



***FOR THE PURPOSES OF INFORMATION ONLY***

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **AL** | Albania | **ES** | Spain | **LS** | Lesotho | **SI** | Slovenia |
| **AM** | Armenia | **FI** | Finland | **LT** | Lithuania | **SK** | Slovakia |
| **AT** | Austria | **FR** | France | **LU** | Luxembourg | **SN** | Senegal |
| **AU** | Australia | **GA** | Gabon | **LV** | Latvia | **SZ** | Swaziland |
| **AZ** | Azerbaijan | **GB** | United Kingdom | **MC** | Monaco | **TD** | Chad |
| **BA** | Bosnia and Herzegovina | **GE** | Georgia | **MD** | Republic of Moldova | **TG** | Togo |
| **BB** | Barbados | **GH** | Ghana | **MG** | Madagascar | **TJ** | Tajikistan |
| **BE** | Belgium | **GN** | Guinea | **MK** | The former Yugoslav | **TM** | Turkmenistan |
| **BF** | Burkina Faso | **GR** | Greece | | Republic of Macedonia | **TR** | Turkey |
| **BG** | Bulgaria | **HU** | Hungary | **ML** | Mali | **TT** | Trinidad and Tobago |
| **BJ** | Benin | **IE** | Ireland | **MN** | Mongolia | **UA** | Ukraine |
| **BR** | Brazil | **IL** | Israel | **MR** | Mauritania | **UG** | Uganda |
| **BY** | Belarus | **IS** | Iceland | **MW** | Malawi | **US** | United States of America |
| **CA** | Canada | **IT** | Italy | **MX** | Mexico | **UZ** | Uzbekistan |
| **CF** | Central African Republic | **JP** | Japan | **NE** | Niger | **VN** | Viet Nam |
| **CG** | Congo | **KE** | Kenya | **NL** | Netherlands | **YU** | Yugoslavia |
| **CH** | Switzerland | **KG** | Kyrgyzstan | **NO** | Norway | **ZW** | Zimbabwe |
| **CI** | Côte d'Ivoire | **KP** | Democratic People's | **NZ** | New Zealand | | |
| **CM** | Cameroon | | Republic of Korea | **PL** | Poland | | |
| **CN** | China | **KR** | Republic of Korea | **PT** | Portugal | | |
| **CU** | Cuba | **KZ** | Kazakstan | **RO** | Romania | | |
| **CZ** | Czech Republic | **LC** | Saint Lucia | **RU** | Russian Federation | | |
| **DE** | Germany | **LI** | Liechtenstein | **SD** | Sudan | | |
| **DK** | Denmark | **LK** | Sri Lanka | **SE** | Sweden | | |
| **EE** | Estonia | **LR** | Liberia | **SG** | Singapore | | |

# LOCATION SYSTEM

This invention relates to a location system and, in particular, to a system which can be used to support personalisation and mobility in typical office environments and in particular to systems of the type disclosed in our British Patent Application No. GB-A-2320089.

It is desirable in a system such as a computer network to have information about location of all the equipment attached to the network and also information about the locations of people using the network and their particular needs at any given time. Usually, the people using the network are mobile but the equipment is fixed. However, in systems such as radio networks, portable equipment can be part of the network and communicates with the network by radio, and even in wired networks equipment can usually be moved around or rotated.

Location systems give absolute information about the locations of objects in space. Typically, the users of information systems are interested in relative location information. This must be derived from the absolute information about objects and persons in space which is the primary data generated by a location system. For example, the sentences "the person is at position *p*" and "the workstation is at position *q*, facing in direction *d*" give absolute location information. However, "the person at position *p* is able to use the workstation at position *q*" gives relative location information. In order to make a location system valuable to the users of information systems such as computer networks, a method of deriving relative location information from absolute location information is required.

A preferred embodiment of the present invention enables relative location information to be derived from absolute location information by expressing the relative location information in terms of spatial containment and overlapping

relationships. These can be calculated using absolute location information. For example, to determine the truth of the sentence "the workstation can be used by the person", a shape *c* is used to represent the area of space in which the workstation can be used, and another shape *p* to represent the person. We can then say that the workstation can be used by the person if, and only if, the space *c* contains the space *p*. By using this absolute location information to fix the locations of *c* and *p*, the truth value of the relative location sentence can then be determined.

In practice, it is necessary to monitor movements of people and objects so that the system can be notified whenever some relative location information becomes true or false. For example, it is necessary to be notified whenever the person *p* moves into or out of the space in which he can use the workstation *c*. If there were, for example, 1000 objects such as workstations attached to a network, each of which has an associated area of space, whenever a person moves there are potentially 1000 relative location statements which could become true or false. To evaluate the truth values of all these statements by calculating the associated containment relationships each time a person or object moves would be too expensive in terms of computing power and a method of indexing the spaces is required so that large numbers of containment relationships can be calculated cheaply.

The invention is defined in the appended claims to which reference should now be made.

A preferred embodiment of the invention will now be described in detail by way of example with reference to the figures in which:

**Figure 1** shows the spaces surrounding a workstation and person represented in a quad tree structure;

**Figure 2** shows a room containing a workstation and person in an embodiment of the invention; and

**Figure 3** shows a flowchart showing the steps performed in an embodiment of the invention.

A preferred embodiment of the present invention operates by using a containment index in a system whereby spaces can be inserted, moved or deleted in a tree structured store. The process of insertion, movement or deletion of spaces is termed an operation on the store. Whenever an operation is performed on the store, four kinds of events are generated. These are as follows:

i) positive containment events which are generated for all pairs of spaces *s* and *s′* where *s* now contains *s′* and did not before the last operation on the store;

ii) negative containment events which are generated for all pairs of spaces *s* and *s′* where *s* does not now contain *s′* and did before the last operation on the store;

iii) positive overlapping events which are generated for all pairs of spaces *s* and *s′* where *s* now overlaps *s′* and did not before the last operation on the store;

iv) negative overlapping events which are generated for all pairs of spaces *s* and *s′* where *s* does not now overlap *s′* and did before the last operation on the store.

The value of using a containment index lies in the fact that an individual operation on the store can be done in time largely independent of the number of spaces in the store whilst still calculating all the changes in value of the containment and overlapping relationships between spaces in the store. Therefore, the time taken to calculate the truth values of the relative statements would be fairly constant whether there were 100, 1000 or 10000 workstations in a system.

The indexing process operates by representing spaces in what is known as a quad tree. This is a known data structure in which a space is broken down into squares approximating the space. Initially, this can be thought of as defining a minimum size of square and representing the shape using these squares and then replacing the small squares with larger squares wherever this is possible so that maximal cover of the space is defined with the smallest number of squares or quad tree cells, as they are known, which covers the whole of the area. This is called maximal cover because it contains the largest cells necessary to cover *s*. Indexing of the spaces can then be based on two theorems about the maximal cover of the spaces.

The maximal cover of a polygon is generated in the following manner.

Suppose we have some suitably large quadtree of extending from a bottom left-hand extreme of (0,0) to a top-right-hand extreme of (n-1,n-1), where n is a suitably large power of 2. The maximal cover of a space is essentially an approximation to the space in terms of quadtree cells. This section describes how to generate the maximal cover of an arbitrary polygon containing only cells in the quadtree which are at least as large as r x r (which is some power of 2 less than 2-to-the-n).

Incidentally, this points to a further benefit of our approach in that we can trade off accuracy against performance by merely varying the value of r - bigger values give smaller numbers of cells in the maximal covers which results in faster performance (but worse accuracy).

First, find the smallest cell, of size bigger than at least r x r, in the quadtree which completely contains the space to be indexed. This can be done by the following method:

1. Find the point (xmin,ymin), where xmin is the smallest value of the x coordinate for any vertex in the sequence of vertices defining the polygon, and similarly ymin

is the smallest value of the y coordinate for any vertex in the sequence of vertices defining the polygon.

2. Find the point (xmax,ymax), where xmax is the greatest value of the x coordinate for any vertex in the sequence of vertices defining the polygon, and similarly ymax is the greatest value of the y coordinate for any vertex in the sequence of vertices defining the polygon.

3. The quadtree cell q0, of size r x r which contains (xmin,ymin) is the square whose bottom left-hand point is at coordinates (xmin*(xmin/r),ymin*(ymin/r)), and whose top right-hand point is at coordinates (xmin*(xmin/r)+r-1,ymin*(ymin/r)+r-1).

4. Now we can perform the following process (expressed in pseudocode) which will set Q to be the smallest quadtree cell which completely contains the space to be covered:

```
Q: = q0
while Q does not contain the point (xmax,ymax) do
     Q: = the parent quadtree cell of Q.
```

Having found a cell in the quadtree which completely contains the space to be indexed, we can generate the smallest set of quadtree cells of size greater than or equal to r which completely covers the given space. This can be done by the following method:

We first define the notions of containment and overlapping for polygons and quadtree cells. A polygon p contains a quadtree cell q if and only if p contains an arbitrary point in q (this can be calculated using a standard algorithm) and no line in the perimeter of p intersects a line in the perimeter of q (this can be calculated using a standard algorithm). A polygon p overlaps a quadtree cell q if and only if some line in the perimeter of p intersects a line in the perimeter of q, or q contains p.

An arbitrary polygon p and quadtree cell q may therefore be related in one of three ways: p contains q, p overlaps q

or p is disjoint from q (this last case obtaining when p neither overlaps nor contains q).

Now call the space to be indexed P and consider a variable Q whose value is initially set to be the minimal cell in the quadtree which completely contains P. We generate the maximal cover of P by recursively splitting up the quadtree cell Q into a set of descendant quadtree cells as follows:

Assume there is a set S which will be built up into the maximal cover of P. This set is initially empty. Note that we know that P overlaps Q initially, because Q contains P.

Note also that each quadtree cell q has 4 children, which we call q0, q1, q2 and q3. We generate the set S using the recursive procedure generate, which is expressed in psuedocode as follows:

```
generate (q: quadtree cell) is

if P contains q then
     S: = S U {q} else if P overlaps q then
     if q is smaller than 2r x 2r
     then
          S: = S U {q} else
          generate(q0)
          generate(q1)
          generate(q2)
          generate(q3)
     end if
end if.
```

The procedure call generate(Q) will terminate with the set S containing all the quadtree cells and only the quadtree cells in the maximal cover of P.

A space, such as the area surrounding a workstation, is shown in Figure 1 represented with maximal cover from a quad tree. In this, the workstation area is divided into three sizes of squares labelled 2, 4, and 6. Not all of the squares are labelled in the Figure. The total space represents the area in which the workstation is useable. A person 8 is shown also formed of quad tree squares. In this example, the person is presently outside the useable area for the workstation,

i.e., the screen region. The indexing method used for finding the relative locations of objects is based on two theorems about the maximal covers of spaces. These are:

i) The containment indexing theorem: this states that a space *s* is contained in a space *t* if, and only if, for each cell in the maximal cover of *s*, there exists exactly one cell in the maximal cover of space *t* which contains *x* or is equal to *x*.

ii) The overlap indexing theorem: spaces *s* and *t* overlap if and only if there exists two cells *x* and *y* in the maximal covers of *s* and *t* respectively such that *x* is equal to *y* or *x* contains *y*, or *y* contains *x*.

Suppose that we have implemented a function called *index* which calculates the maximal cover of any space. To index spaces in a tree, we maintain a representation of the quadtree where each node is annotated with the set of indexed spaces whose index contains that node. To index a space *s* we calculate the nodes in *index s* and add *s* to the annotation of each node. The function *indexed* returns the set of indexed spaces for any node in the quadtree, and the function *index_count* returns the number of cells in the index of a space (i.e., *index_count=#(index s))*.

From the containment indexing theorem, we can see that given spaces *t* and *s*, if *s* is contained in *t* then each cell in *index s* is a descendant of exactly one cell in *index t;* so, if we search each of the trees rooted at a cell of *index t* we should visit each member of *index s* exactly once.

If we maintain an associative array *count* to count the number of times we visit a space we can calculate the set of "contained" spaces contained by *t* by searching each of the trees rooted at a cell of *index t* and for each cell *y* that we visit, incrementing the value of *count[s]* for each space *s* in *indexed y*, and returning those spaces *s* for which *count[s]* =

*index_count s*. The algorithm can be expressed in this pseudocode:

```
count:=Ø
contained:=Ø
foreach x ∈ index t
    foreach y ∈ descendants x
        foreach s ∈ indexed y
            count[s]:=count[s]+1
            if count[s]=index_count s then
                contained:=contained ∪ {s}
```

We can use a similar algorithm to calculate the set of "contained" spaces which contain the space *t*, this time by touring the ancestors of *t*:

```
count:=Ø
containers:=Ø
foreach x ∈ index t
    foreach y ∈ ancestors x
        foreach s ∈ indexed y
            count[s]:=count[s]+1
            if count[s]=index_count t then
                containers:=containers ∪ {s}
```

From the overlap indexing theorem, we can see that given spaces *t* and *s*, if *s* overlaps *t* then some cell in *index s* is either a descendant or an ancestor of some cell in *index t*; so, if we search each of the trees rooted at a cell of *index t* and all of the ancestors of *index t* we should visit some member of *index s*. We can express the algorithm in this pseudocode:

```
overlapped:=Ø
foreach x ∈ index t
    foreach y ∈ ancestors x ∪ descendants x
        foreach s ∈ indexed y
            overlapped:=overlapped ∪ {s}.
```

An optimisation is to calculate all three values as part of the insertion and deletion process. The changes in the values when a space is moved can be calculated by deleting the space at the old position and inserting the space at the new position. The deletion operation generates the sets *contained*, *containers* and *overlapped*; the insertion operation generates the sets *contained'*, *containers'* and *overlapped'*.

So when a space s moves, the positive and negative containment and overlapping events can be generated by evaluating the following expressions:

contained - (contained *n* contained') is the set of all spaces which have stopped being contained by s;

contained' - (contained *n* contained'" is the set of all spaces which have just become contained by s;

containers - (containers *n* containers') is the set of all spaces which have stopped being containers of s;

containers' - (containers *n* containers') is the set of all spaces which have just become containers of s;

overlapped - (overlapped *n* overlapped') is the set of all spaces which have stopped overlapping s;

overlapped' - (overlapped *n* overlapped') is the set of all spaces which have just started overlapping s.

Figure 2 shows physically how the system might operate.. It represents the inside of a room 10 which contains a workstation 12 and a person 14. Fitted to either side of the workstation 12 are position indicators 16 which send signals identifying uniquely each position indicator. A position indicator is also attached to the person 14.

Signals transmitted by the position indicator 16 are detected by a number of sensors 18 located in the room. These may be ultrasonic detectors which interrogate indicators on objects and people at regular intervals. Position is computed trigonometrically from the time taken for signals to reach the sensors 18. Signals detected can then be sent back to a control unit 20 associated with the computer network. Thus, the absolute location information of the indicator 16 is derived from the sensors 18 by the control unit 20 which is able to use this for detecting whether or not e.g., the person is in an area which enables him sitting in it to use the workstation 12.

The control unit will preferably poll the sensors 18 to transmit interrogation signals one at a time to the position

indicators. Each position indicator in the room will respond by sending a response back to the sensor. The distance of the object from the sensor can then be determined from the total time taken for a response to arrive after the interrogation signal has been transmitted. The interrogation signal may be a broadcast signal to all position indicators. However, this may lead to contention in the responses. It is therefore preferably directed to each position indicator one at a time using an identifier for that position indicator.

To get an accurate position signal, responses are required at three sensors at different locations.

Positions can be determined for any number of position indicators and may be determined in either two or three dimensions.

The position indicators 16 are also carried by people moving around in the building between rooms such as room 10 which have workstations or other equipment connected to the computer network located within them. Typically, these sensors on people will be interrogated more regularly than the sensors on workstations.

At the control unit 20, the information from the position sensors is used to represent the workstation space and a person space and the space associated with any person in the room 10 in a form similar to that shown in Figure 1.

These typed containment events can then be used by the computer network to deliver application software as necessary. For example, a software application can be given a designation called "Follow Me". The user of that application can then move from workstation to workstation throughout a building and when he becomes close to or moves into the accessible area of a particular workstation, the application will be loaded to that workstation and he can continue to use the application. The callback registration storage and lockup step 36 uses the typed containment events and contacts the appropriate applications with the relevant relative location information,

thereby enabling the system to support mobile personalised software applications in box 38.

This describes a test system in which the indexing system is used to provide "Follow Me" X computer sessions.

A user has an X Window System desktop running on an XRFB server. RFB stands for "remote frame buffer". The XRFB server sends low-level display information to a remote machine running an RFB display server and receives keyboard and mouse input from that machine. This enables a person to use that remote machine as though it is running their X desktop locally. There is a control interface on the XRFB server which enables an application to change the remote display server to which the XRFB server sends low-level display information and from which it receives keyboard input; using this interface an application can control where the physical view of an X desktop is located.

There are several machines running RFB display servers. Each of these machines is tracked by an ultrasonic location system of the type discussed above which generates a stream of location and orientation events for each of the machines. For each machine these events are translated into a stream of location and orientation events for a space which represents the space around the machine in which the screen is readable. A person is also tracked by the ultrasonic location system, which generates a stream of location and orientation events for the person. These events are translated into a stream of location and orientation events for a small space which contains the person. Whenever any of these spatial location and orientation events occurs, the corresponding space is indexed (reindexed) in the spatial index at its new position.

The application which provides mobile X desktops registers for a callback whenever the area around a person becomes completely contained by the area in which a machine's screen is visible. When this happens, the Follow Me application is called back by the spatial index with the information about which machine's screen area the person has

moved into. The application sends a command to the XRFB server to send display output to, and receive keyboard and mouse input from, the machine whose screen area the person has moved into.

The application which provides mobile X desktops also registers for a callback whenever the area around a person ceases to overlap the area in which a machine's screen is visible. When this happens, the Follow Me application is called back by the spatial index with the information about which machine's screen area the person has ceased to overlap. The application sends a command to the XRFB server to stop sending display output to, and receiving keyboard and mouse input from, the machine whose screen area the person has moved away from. (The reason for using positive containment and negative overlapping events is that this provides a degree of hysteresis, thus avoiding undesirable behaviour when a person stands on the boundary of a display area).

Using this system we can see that as a person walks up to a machine, his X desktop will physically appear on that machine and as he walks away his desktop will disappear again.

The indexing system could be used in any application which needs to evaluate spatial containment and overlapping relations in real time. Apart from support for mobile applications, which we have described here, other possible applications involve navigating remote-controlled vehicles around complex environments and any other systems which would benefit from this type of evaluation.

Other example applications involve spaces which are not in the real world such as support for detecting when objects stand in some relation to each other in a shared virtual environment or a video game (e.g., performing collision detection for complex shapes).

Thus, it will be appreciated that a location system of the type described above can be extremely useful in network systems where it is desirable for people to have access to the network from more than one location. The ability of the

embodiment of the invention to load up personalised application software onto a terminal will lead to a significant reduction in logging on time to workstations. Clearly, when a person comes into the useable area, the system can be configured to automatically log-in as that person. Furthermore, when he moves away, it can be made to automatically log-out, thus improving on the security problems associated with conventional workstations where users have to physically log-in and log-out and a user who does not log-out but moves away from his workstation leaves possibly confidential information available to other persons.

**CLAIMS**:

1. A method for locating objects which carry location devices comprising the steps of:

a) repeatedly interrogating the location devices on each object to provide position signals;

b) detecting position signals provided by the location devices;

c) determining from the position signals the position of each object and an area associated therewith;

d) storing position and area data for each object;

e) determining the relative locations of the thus determined objects and associated areas.

2. A method according to claim 1 in which at least some of the objects have at least two location devices and including the step of determining the orientation of such objects.

3. A method according to claim 1 in which the step of determining the relative location of users comprises determining whether an object has moved since it was last interrogated and, if it has moved, determining whether it now overlaps or is within an area associated with another object.

4. A method according to claim 3 further comprising determining whether an object which previously overlapped or was contained within an area associated with another object now falls outside that area.

5. A method according to any preceding claim in which some of the objects are computer terminals and other objects are people.

6. A location system for objects which carry location devices comprising:

a) means for repeatedly interrogating the location devices on each object to provide position signals;

b) means to detect the position signals;

c) means for determining from the position signals the position of an object and an area associated with each object from the position signals;

d) means for storing position and area data for each object;

e) means for determining the relative location of the thus determined objects and associated areas.

7. A location system according to claim 6 in which at least some of the objects have at least two location devices and the system includes means for determining the orientation of such objects.

8. A location system according to claim 6 or 7 in which the means for determining the relative locations of areas comprises means for determining whether the object has moved since it was last interrogated, and means for determining whether its area overlaps or is within an area associated with any other object.

9. A location system according to claim 8 comprising means for determining whether an object that previously overlapped or was contained within an area associated with another object now falls outside that area.

10. A location system according to any of claims 6 to 9 in which some of the objects are computer terminals and other objects are people.

11. A location system for objects comprising:

means for determining the absolute locations of the objects;

means for deriving spatial data associated with the objects; and

means for deriving relative location data from the spatial data.

12. A location system according to claim 11 in which the means for determining the absolute locations of objects comprises a position indication system including sensors carried by objects.

13. A location system for objects comprising:

a) a position indication system for providing position signals for location devices carried by objects;

b) means for determining from the position signals the position of an object and an area associated therewith;

c) means for storing position and area data for each object; and

e) means for deriving the relative locations of the thus determined objects and associated areas.

Figure 1

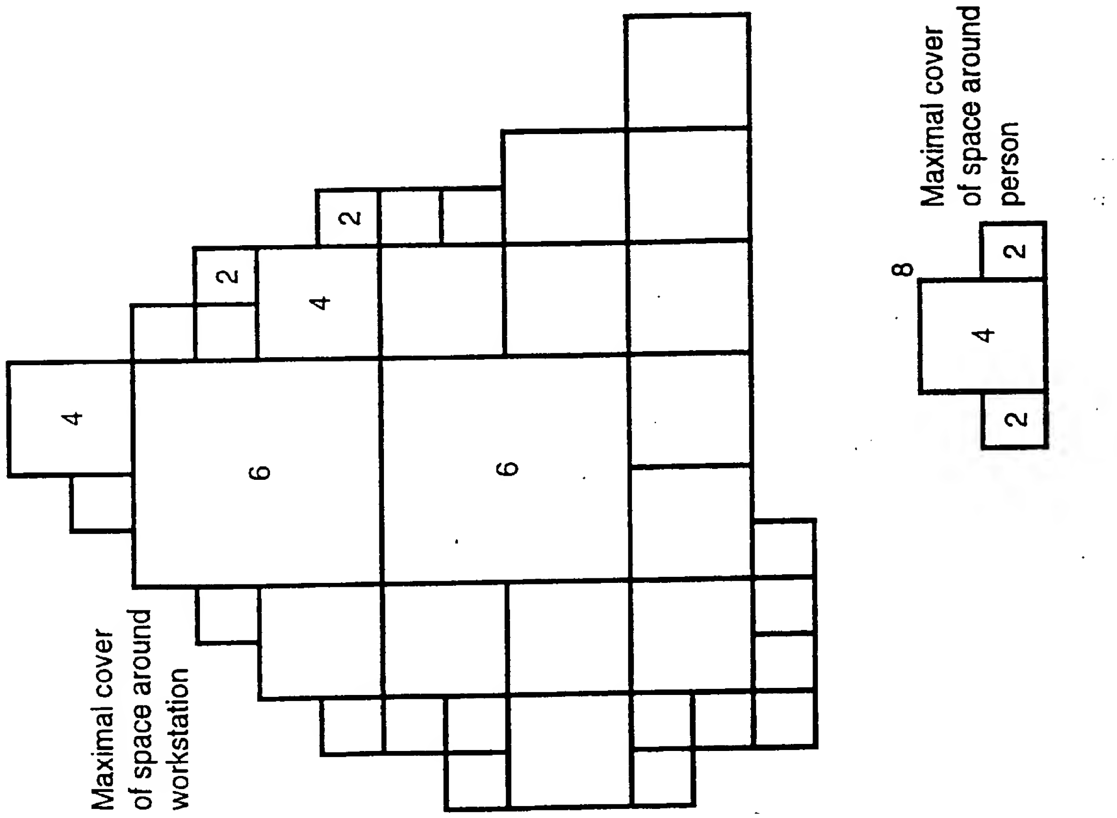
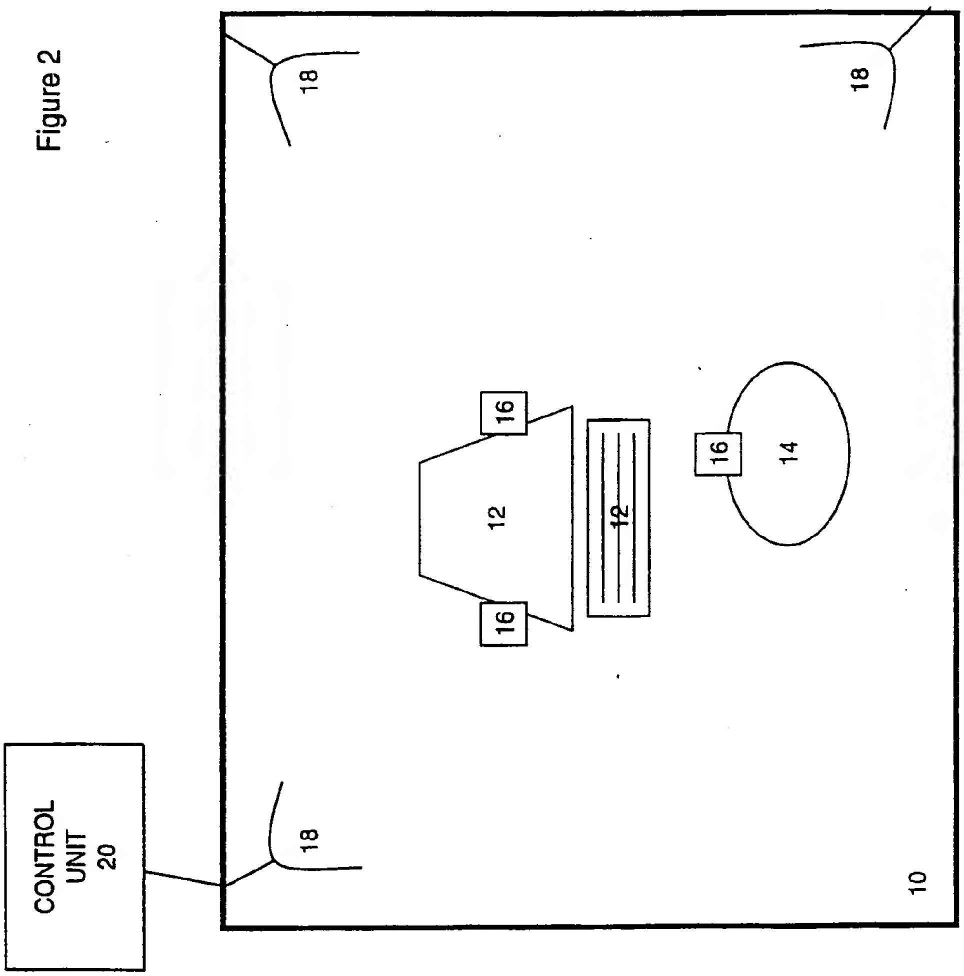

Maximal cover of space around workstation
4
6
6
2
4
2
Maximal cover of space around person
8
4
2
2

Figure 2



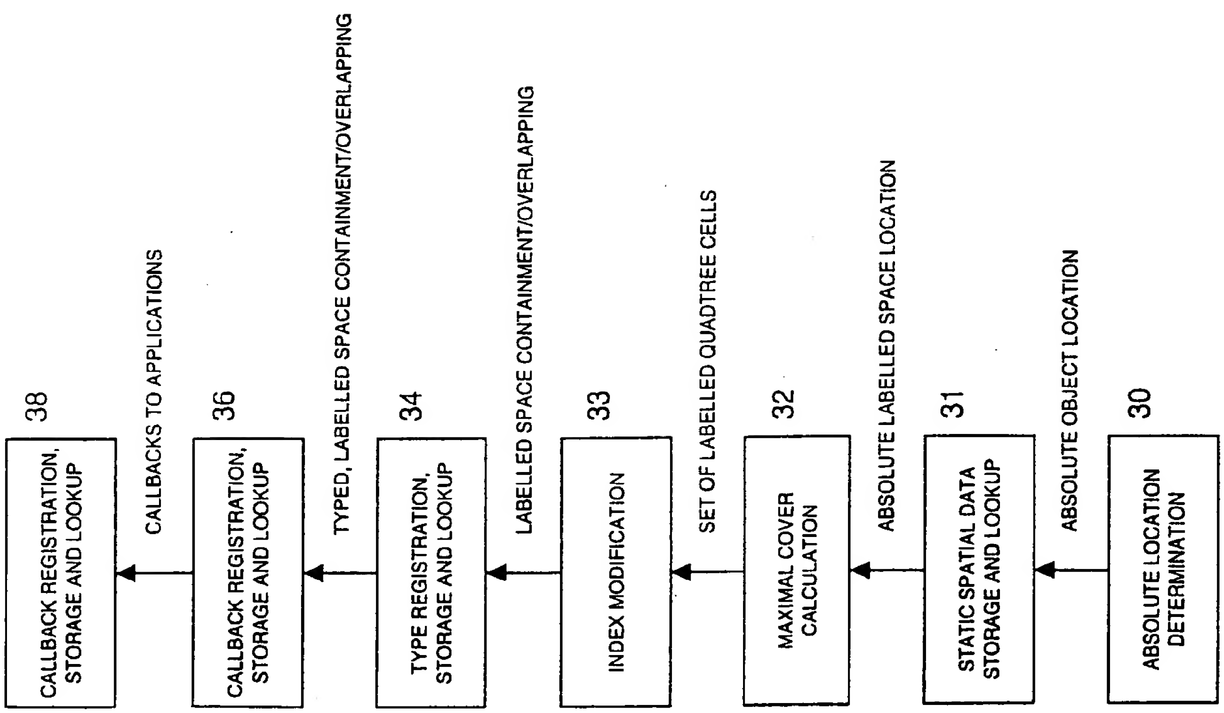

30
ABSOLUTE LOCATION DETERMINATION
ABSOLUTE OBJECT LOCATION
31
STATIC SPATIAL DATA STORAGE AND LOOKUP
ABSOLUTE LABELLED SPACE LOCATION
32
MAXIMAL COVER CALCULATION
SET OF LABELLED QUADTREE CELLS
33
INDEX MODIFICATION
LABELLED SPACE CONTAINMENT/OVERLAPPING
34
TYPE REGISTRATION, STORAGE AND LOOKUP
TYPED, LABELLED SPACE CONTAINMENT/OVERLAPPING
36
CALLBACK REGISTRATION, STORAGE AND LOOKUP
CALLBACKS TO APPLICATIONS
38
CALLBACK REGISTRATION, STORAGE AND LOOKUP

Figure 3

# INTERNATIONAL SEARCH REPORT

International Application No
PCT/GB 99/01858

**A. CLASSIFICATION OF SUBJECT MATTER**
IPC 6 G01S5/18 G01S5/30 G01S15/87

According to International Patent Classification (IPC) or to both national classification and IPC

**B. FIELDS SEARCHED**

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 G01S

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

| Category ° | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| X | GB 2 298 098 A (TAGWARE LTD) 21 August 1996 (1996-08-21) | 1,2,5-7, 10,13 |
| A | page 1<br>page 4 - page 11 | 3,4,8,9, 11 |
| X | WO 97 14048 A (AMERICAN TECHNOLOGY CORP) 17 April 1997 (1997-04-17) | 11,12 |
| A | abstract<br>page 6, line 16 - page 8, line 24<br>figures 5,6,8-10 | 1,6,13 |

-/--

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

| Date of the actual completion of the international search | Date of mailing of the international search report |
|---|---|
| 10 September 1999 | 24/09/1999 |

| Name and mailing address of the ISA | Authorized officer |
|---|---|
| European Patent Office, P.B. 5818 Patentlaan 2<br>NL - 2280 HV Rijswijk<br>Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,<br>Fax: (+31-70) 340-3016 | Roost, J |

2

Inter• nal Application No
PCT/GB 99/01858

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

| Category ° | Citation of document, with indication,where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| X | GB 2 320 089 A (OLIVETTI RES LTD ;ORACLE CORP (US)) 10 June 1998 (1998-06-10) cited in the application | 11,12 |
| A | abstract<br>page 1<br>page 3 - page 4<br>page 12<br>page 41, line 1 - page 42, paragraph 4<br>figures 1,9,10 | 1,2,5-7, 10,13 |
| A | GB 2 265 038 A (OLIVETTI RES LTD ;DIGITAL EQUIPMENT CORP (US))<br>15 September 1993 (1993-09-15)<br>page 16, paragraph 5 - page 21, paragraph 1<br>figures 5,6 | 1,6,11, 13 |
| A | WO 95 27963 A (LEW HELEN ;MINTZ YOSEF (IL)) 19 October 1995 (1995-10-19)<br>page 9, line 35 - page 11, line 5 | 3,4 |
| A | EP 0 766 098 A (SQUARE CO LTD)<br>2 April 1997 (1997-04-02)<br>abstract; figures 1,2 | 1,6,11, 13 |
| A | WO 97 01154 A (MEDESTATE LTD ;MITCHESON MARK (GB); KEVAN ROBERT ANDREW (GB); GREN)<br>9 January 1997 (1997-01-09)<br>page 1, paragraph 3 - page 4, last paragraph<br>figures 1,2 | 1,6,11, 13 |

2

# INTERNATIONAL SEARCH REPORT

ormation on patent family members

Inter nal Application No
PCT/GB 99/01858

| Patent document cited in search report | | Publication date | Patent family member(s) | | | Publication date |
|---|---|---|---|---|---|---|
| GB 2298098 | A | 21-08-1996 | NONE | | | |
| WO 9714048 | A | 17-04-1997 | US | 5781150 | A | 14-07-1998 |
| | | | AU | 7438696 | A | 30-04-1997 |
| | | | EP | 0864104 | A | 16-09-1998 |
| GB 2320089 | A | 10-06-1998 | WO | 9828634 | A | 02-07-1998 |
| GB 2265038 | A | 15-09-1993 | AU | 663770 | B | 19-10-1995 |
| | | | CA | 2131726 | A | 16-09-1993 |
| | | | FI | 944111 | A | 02-11-1994 |
| | | | WO | 9318476 | A | 16-09-1993 |
| | | | HU | 70721 | A | 30-10-1995 |
| | | | JP | 7504545 | T | 18-05-1995 |
| WO 9527963 | A | 19-10-1995 | AU | 703874 | B | 01-04-1999 |
| | | | AU | 2216295 | A | 30-10-1995 |
| | | | EP | 0758476 | A | 19-02-1997 |
| EP 0766098 | A | 02-04-1997 | JP | 9096672 | A | 08-04-1997 |
| | | | CA | 2186200 | A | 30-03-1997 |
| WO 9701154 | A | 09-01-1997 | NONE | | | |